A RECONFIGURABLE SIGNAL PROCESSING IC WITH AN EMBEDDED FLASH MEMORY DEVICE

Field of the Invention

[0001] The present invention relates to dynamically reconfigurable processing units connected to a Flash EEPROM memory subsystem. More specifically, the present invention relates to a reconfigurable signal processing integrated circuit with an embedded Flash memory device for non-volatile storage of code, data and bit-streams. The unit is integrated into a single chip together with a microprocessor core.

Background of the Invention

[0002] Increasing complexity of system design and shorter time-to-market requirements are leading research towards the investigation of hybrid systems including processors enhanced by programmable logic as readily known by those skilled in this technical field. In this respect, reference is made to the work by Young-Don Bae et al., "A Single-Chip Programmable Platform Base on A Multithreaded Processor and Configurable Logic Clusters", ISSCC 2002 Digest of Technical Papers, pp 336-337, Feb. 2002. Moreover, a further reference which may be considered is an article by Zhang et al., "A 1V Heterogeneous Reconfigurable Processor IC for Baseband Wireless Applications", ISSCC 2000 Digest of Technical Papers, pp 68-69,488, Feb. 2000.

[0003] At the same time increased costs of mask sets and a shorter time-to-market available for new products are leading to the introduction of systems with a higher degree of programmability and configurability, such as system-on-chips with configurable processors, embedded FPGA and embedded flash memory. Moreover, the availability of an advanced embedded flash technology, based on a NOR architecture, together with innovative IP's, like embedded flash macrocells with special features, is a key factor.

[0004] For a better understanding of the present invention reference is also made to the Field Programmable Gate Array (FPGA) technology combining standard processors with embedded FPGA devices. These approaches allow one to configure into the FPGA at deployment time exactly the required peripherals, exploiting temporal reuse by dynamically reconfiguring the instruction set at run time based on the currently executed algorithm.

[0005] The existing models for designing FPGA/processor interaction can be grouped in two main categories: 1) the FPGA is a co-processor communicating with the main processor through a system bus or a specific I/O channel; and 2) the FPGA is described as a function unit of the processor pipeline. The first group includes the GARP processor, known from the article by T. Callahan, J. Hauser, and J. Wawrzynek, "The Garp Architecture And C Compiler" IEEE Computer, 33(4): 62-69, April 2000. A similar architecture is provided by the A-EPIC processor that is disclosed in the article by S. Palem and S. Talla, "Adaptive Explicit Parallel Instruction Computing", Proceedings

of the fourth Australian Computer Architecture Conference (ACOAC), January 2001.

[0006] In both cases the FPGA is addressed via dedicated instructions, moving data explicitly to and from the processor. Control hardware is kept to a minimum since no interlocks are needed to avoid hazards, but a significant overhead in clock cycles is required to implement communication. When the number of cycles per execution of the FPGA is relatively high, the communication overhead may be considered negligible.

[0007] In the commercial world, FPGA suppliers such as Altera Corporation offer digital architectures based on U.S. Patent No. 5,968,161 to T.J. Southgate, "FPGA Based Configurable CPU Additionally Including Second Programmable Section For Implementation Of Custom Hardware Support". Other suppliers (Xilinx, Triscend) offer chips containing a processor embedded on the same silicon IC with embedded FPGA logic. See for instance U.S. Patent 6,467,009 to S.P. Winegarden et al., "Configurable Processor System Unit", assigned to Triscend Corporation.

[0008] However, those chips are generally loosely coupled by a high speed dedicated bus, performing as two separate execution units rather than being merged in a single architectural entity. In this manner the FPGA does not have direct access to the processor memory subsystem, which is one of the strengths of academic approaches outlined above. In the second category (FPGA as a function unit) we find architectures commercially known as PRISC, Chimaera and ConCISe.

[0009] In all these models, data is read and written directly on the processor register file minimizing overhead due to communication. In most cases, to minimize control logic and hazard handling and to fit in the processor pipeline stages, the FPGA is limited to combinatorial logic only. This severely limits the performance boost that can be achieved.

[0010] These approaches represent a significant step toward a low-overhead interface between the two entities. Nevertheless, due to the granularity of FPGA operations and its hardware oriented structure, their approach is still very coarse-grained, reducing the possible resource usage in parallel and again including hardware issues not familiar nor friendly to software compilation tools and algorithm developers.

[0011] Thus, a relevant drawback in this approach is often the memory data access bottleneck that often forces long stalls on the FPGA device when fetching on the shared registers enough data to justify its activation.

Summary of the Invention

[0012] An object of the present invention is to provide a reconfigurable processing unit that is connected to a memory architecture having functional and structural features offering significant performance and power consumption enhancements with respect to a traditional signal processing device.

[0013] The invention overcomes the limitations of similar preceding architectures relying on an embedded device of a different nature, and this is based upon a new approach to processor/memory interface. According

to a first embodiment of the present invention, the reconfigurable processing unit targets image-voice processing and recognition application domains by joining a configurable and extensible processor core and an SRAM-based embedded FPGA.

[0014] More specifically, the processing unit according to the invention may includes an S-RAM based embedded FPGA unit structured for FPGA reconfigurations having a specific programming interface connected to a port FA of the Flash memory device through a direct memory access (DMA) channel.

Brief Description of the Drawings

[0015] The features and advantages of the processing unit according to the present invention will become apparent from the following description of a best mode for carrying out the invention given by way of a non-limiting example with reference to the enclosed drawings.

[0016] Figure 1 is a block diagram of a processing unit architecture for data processing according to the present invention;

[0017] Figure 2 is a block diagram of a Flash memory architecture embedded into the processing unit of Figure 1;

[0018] Figure 3 is a schematic view of a system memory hierarchy provided by the present invention;

[0019] Figure 4 is a block diagram of a specific processor extension according to the present invention with respect to added DSP instruction examples;

[0020] Figure 5 is a block diagram of a further specific processor extension according to the present

invention with respect to an optimized fixed-point calculation of the square root accounts;

[0021] Figure 6 is a table view showing the overall performance improvements for a face recognition task implemented by the processing unit of the present invention; and

[0022] Figure 7 is a schematic chip micrograph according to the present invention.

Detailed Description of the Preferred Embodiments

[0023] With reference to the drawings, generally shown at 1 is a processing unit formed according to the present invention for digital signal processing based on reconfigurable computing. The processing unit 1 includes an embedded Flash memory 4 for non-volatile storage of code, data and bit-streams, and an additional S-RAM based embedded FPGA 3 formed for the configuration purposes of the present invention. More specifically, an 8Mb application-specific embedded flash memory 4 is disclosed. The memory 4 is integrated into a single chip together with a microprocessor 2 and the FPGA structure 3.

[0024] Advantageously, application-specific hardware units are added and dynamically modified by the embedded FPGA 3 reconfiguration. By implementing application-specific vector processing instructions the processing unit 1 shows a peak computing power of 1GOPS.

[0025] Efficient read-write-erase access to code, data and FPGA bitstreams is provided by the Flash memory 4 based on a modular 8Mb, 4-bank Flash memory, as will be more clearly explained below.

[0026] The processing unit 1 comprises three content-specific I/O ports and delivers an aggregate peak read throughput of 1.2GB/s. The system architecture 1 is illustrated in Figure 1. The functional purposes of the embedded FPGA 3 are as follows: i) extension of the processor datapath supporting a set of additional special purpose C-callable microprocessor instructions; ii) bus-mapped coprocessors connected to the system bus through a master/slave interface; and iii) flexible I/O to connect external units or sensors with application-specific communication protocols.

[0027] Even though such different circuit purposes would require different kinds of programmable logic for best implementation of either arithmetic-dominated or control-dominated logic, a single programmable logic subsystem 3 has been implemented to be shared among different purposes both in space (same configuration) and time (subsequent configurations).

[0028] The single, high I/O count, fine-grain e-FPGA 3 operates as a datapath for the microprocessor pipeline and as dedicated control logic for bus coprocessor and I/O control interface. The FPGA has a specific programming interface 7 connected to a port FP of said Flash memory device 4 through a DMA channel 8. FPGA reconfiguration is concurrent with software execution.

[0029] A local bus 6 connects a dedicated 32-bit Flash memory port FP to the FPGA programming interface 7. A DMA channel 8 handles the bitstream transfer while the microprocessor 2 fetches instructions and data from different Flash memory ports: 64-bit wide code port

(CP) and data port (DP). To support streaming applications a 1kB dual-port buffer 9 is used to interface fast decoding hardware and slower software running on the processor 2. The memory sub-system architecture is shown in Figure 2.

[0030] The modular structure of the memory (dotted line) includes: charge pumps 10 (Power Block), testability circuits 11 (DFT), a power management arbiter 12 (PMA), and a customizable array 13 of N independent 2Mb flash memory modules 16. Depending on the storage requirements the number N may be chosen with N=4 in the current implementation.

[0031] The modular memory features (N+2) 128-bit target ports and implements an N-bank uniform memory 13. As previously mentioned, three content-specific ports are dedicated to code (CP, 64-bit wide), data (DP, 64-bit) and FPGA bit stream configurations (FP, 32-bit). A 128 bit sub-system crossbar 15 connects all the architecture blocks and the eight bit microprocessor 2.

[0032] The main features of the flash memory device 4 includes a charge pump 10 sharing among different flash memory modules 16 through the PMA arbiter 12 in a multi-bank fashion. Moreover, the use of a small eight bit microprocessor 2 allows easy memory system test and adds complex functionalities for data management, and the use of an ADC (Analog-to-Digital Converter), required by the application, increase system self-test capability.

[0033] The third FP port of the Flash device 4 is dedicated to manage embedded-FPGA (e-FPGA) configurations data stored in flash memory modules. The

FP port is read-only and provides fast sequential access for bit stream downloading. The FP has four configuration registers replicating the information stored in CP port that must be used to write e-FPGA configurations data.

[0034] The output data word bus and the address bus are 32 bits wide. The FP port uses a chip select to access in the addressable memory space, and a burst enable to allow burst serial access. In a read operation, an output ready signal is tied low when data is not immediately available, so that it can act as a wait state signal.

[0035] The eight-bit microprocessor 2 (uP) performs additional complex functions (defragmentation, compression, virtual erase, etc.) not natively supported by the DP port, and assists for built-in self-test of the memory system. The (N+2)x4 128-bit crossbar 15 connects the modular memory with the four initiators (CP, DP, FP and uP) providing that at least three flash memory modules 16 can be read in parallel at full speed.

[0036] The memory space of the four modules 16 is arranged in three programmable user-defined partitions, each one devoted to a port. The memory system clock can run up to 100MHz, and reading three modules 16 with a 128bit data bus and 40ns access time results in a peak read throughput of 1.2GB/s. Each 2Mb flash memory module 16 has a 128-bit IO data bus with 40ns access time, resulting in 400Mbyte/s, and a program/erase control unit. Simultaneous memory operations use the power management arbiter 12 (PMA) for optimal scheduling.

[0037] Available power and user-defined priorities are considered to schedule conflicting resource requests in a single clock cycle. The memory device 4 allows up to four simultaneous operations, with a limit of one both for write and erase.

[0038] Figure 3 depicts the memory hierarchy and its parallel architecture across the processing unit 1. The ports CP and DP are interfaced to the 64-bit, 800MB/s AHB system bus 6. At a system clock rate of 100MHz each I/O port can independently operate at maximum speed. An aggregate peak read rate of 1.2GB/s can thus be sustained as it is limited by memory access time. In the current implementation, the e-FPGA reconfiguration takes 500µs at 100 MHz. 50MB/s average throughput out of the available 400MB/s are currently sustained by the e-FPGA configuration interface 7.

[0039] System performance was evaluated for an image processing application (facial recognition) and a speech recognition application. More than 20 specific instructions were designed as C/assembly-callable functions, automatically translated to RTL, then synthesized and mapped to the e-FPGA.

[0040] Figures 4 and 5 show two examples of specific microprocessor extensions. Figure 4 relates to an eight-issue, eight-bit, L2 calculation accounts for 23 eight-bit arithmetic operations and six 64-bit operations requiring about 10k ASIC equivalent gates.

[0041] Figures 5 relates to a datapath for an optimized fixed-point calculation of the square root accounts for twelve 32-bit operations for about 2k ASIC equivalent gates. The overall performance improvements for the face recognition tasks are shown in Figure 6.

[0042] Execution time is compared for a 32-bit RISC with basic DSP extensions (MAC, zero-overhead loops, etc) and the same processor enhanced with application-specific instructions. Measured speed-ups range from 1.8x to 10.6x (on the most-demanding task), with an overall improvement of 8.5x. Switching between algorithm stages requires only one reconfiguration of the e-FPGA. Reconfiguration time is negligible.

[0043] The speed-up factors take into account the possible multi-cycle clock penalty due to processor-FPGA synchronization in case of instruction extensions slower than the processor clock. Energy efficiency figures are reported in Figure 6 also.

[0044] As the average power consumption of the system extended with the e-FPGA is slightly higher (10-15%), the energy reduction for executing each of the tasks on its specific HW configuration (power-delay product improvement) results in an overall reduction of 6.7x. Only one task showed slightly worse total execution energy, though showing benefits on execution speed.

[0045] The last column of Figure 6 reports the power-delay improvement of each specific HW configuration compared to the general-purpose counterpart. Energy required for e-FPGA reconfiguration is always negligible. Measurements show the best energy efficiency in the range of several MOPS/mW at 1.8V supply. It lies between conventional ASIP/DSP and dedicated configurable hardware implementations.

[0046] The full-processing unit on a single chip is implemented in a 0.18μm, 2PL-6ML CMOS embedded Flash technology. Chip area is 70mm^2, and the technology and

device characteristics are summarized in Figure 6. A chip micrograph is shown in Figure 7.